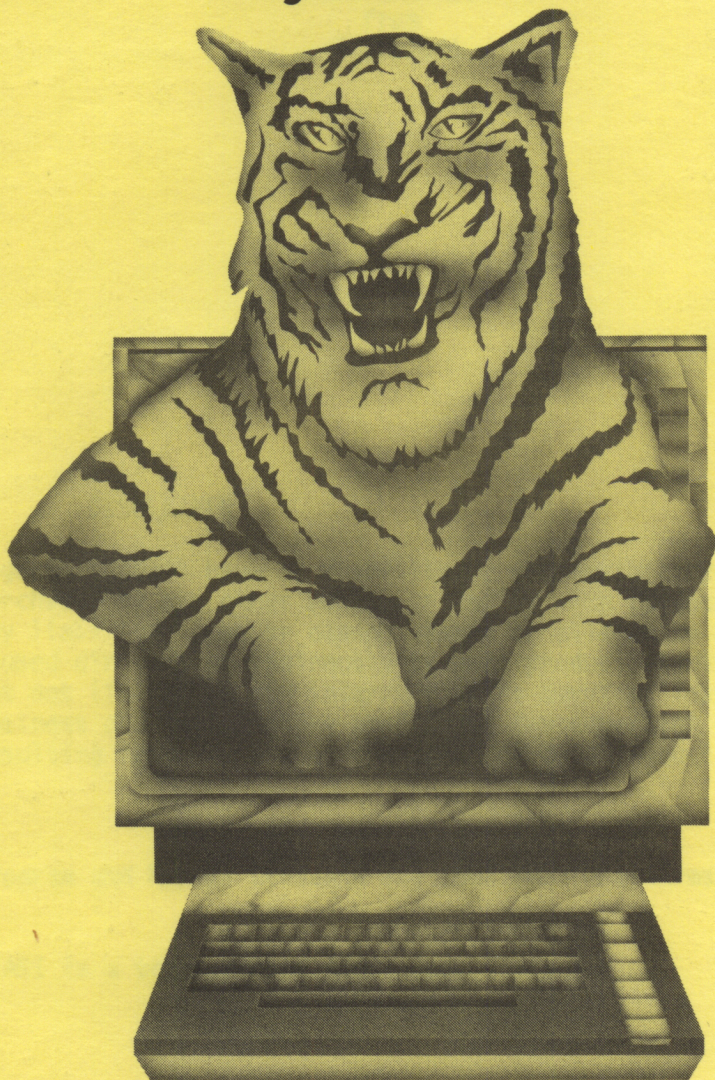


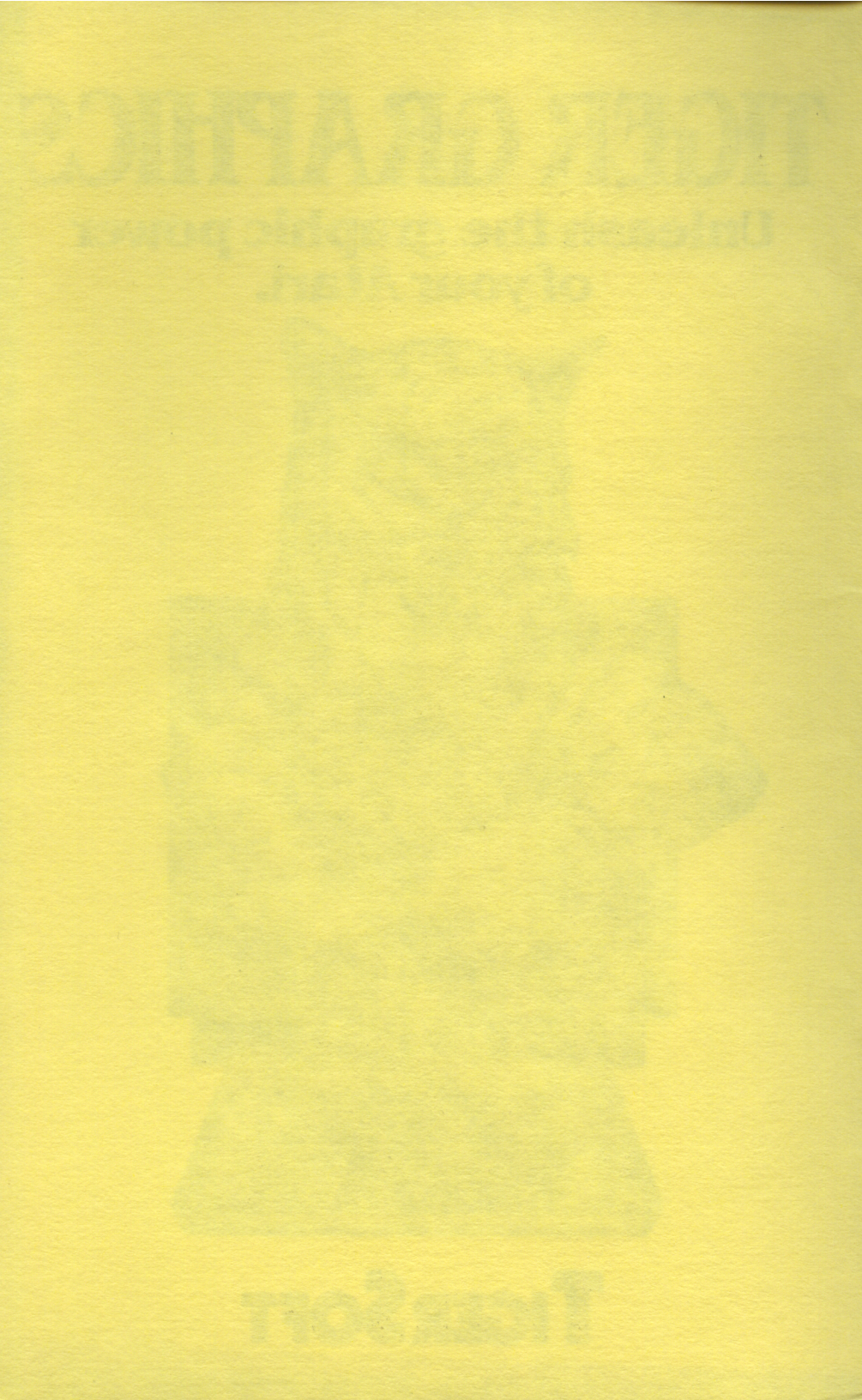
# TIGER GRAPHICS

Unleash the graphic power  
of your Atari.



# TIGERSOFT







TIGER GRAPHICS  
from  
TigerSoft

DISCLAIMER

TigerSoft shall have no liability or responsibility to the purchaser or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by this product, including but not limited to any interruption in service, loss of business and anticipatory profits or consequential damages resulting from the use or operation of this product.

Program and Manual Contents (C) 1983 Mark Alan Armstrong

ATARI is a trademark of ATARI Inc.







## REQUIRED EQUIPMENT

Tiger Graphics requires 48K RAM, and a disk drive.

## CREATING BACK-UP DISK

TigerSoft allows the purchaser of this product, Tiger Graphics, to create a back-up copy of this program. You are prohibited from reproducing for the purpose of distributing this software. To make a back-up, remove any cartridge, insert Tiger Graphics, and turn on the computer. Type:

**MAKEBACK**

The back-up should be a blank formatted disk.



## TABLE OF CONTENTS

	PAGE
INTRODUCTION	1
" HOW TO "	
" HOW TO " DEFINE A PICTURE	2
" HOW TO " DRAW A PICTURE	5
" HOW TO " SAVE A PICTURE	15
HARDWARE and PROGRAM DESCRIPTION	
DISPLAY LIST	17
GRAPHICS MODES	17
CHARACTER SETS ( FONTS )	21
INTERRUPTS	23
SCROLLING	24
SHADOW REGISTERS	25
MODULES	
KERNAL COMMANDS	26
DEF-PIC COMMANDS	30
DRAW-PIC COMMANDS	35
DISK COMMANDS	43
GENERAL INFORMATION	
GRAPHIC REGISTERS	45
VERTICAL BLANK ADDRESSES	50
HELPFUL HINTS	52
MEMORY MAP	53
GRAPHIC MODE TABLE	54

## INTRODUCTION

Tiger Graphics was designed to bridge the gap between graphic and character generators. The main feature of Tiger Graphics is its ability to generate character sets (fonts), thus eliminating the need to design and place each character. Tiger Graphics can generate up to sixteen fonts, or 1024 different characters. Most popular games use character graphics. The memory efficiency of character sets allows for large scrolling pictures. Tiger Graphics can also use bit map modes.

This program does not attempt to isolate the user from the hardware. There are no magical "HI RES" commands available. Tiger Graphics brings the full power of the hardware to the user in easy commands. This program is also excellent for the beginner who wants to learn about the hardware. An example of this is the display list interrupt. No knowledge of assembly language, or how a interrupt works, is needed with Tiger Graphics. Simple commands will insert the interrupt anywhere on the display. This manual explains the hardware, but not in great detail. It does not replace DE RE ATARI or the hardware manual.

The manual is divided into three sections. First read the "HOW TO" section, which does not attempt to describe but shows how to use Tiger Graphics. The second section covers the hardware, and how the program works. The last section describes all the commands used.

## " HOW TO " DEFINE A PICTURE

Insert the Tiger Graphics disk and turn on the computer. A menu should appear listing the modules. The program is divided into 3 modules. First we have to define a picture using the DEF-PIC module. To do so, type:

DEF-PIC

A menu should appear listing the DEF-PIC commands. Tiger Graphics allows you to use all the available graphics modes. To display the graphics modes, type:

MODES

A table of the modes should appear. To define a picture using character mode CHAR4, type:

DCHAR4

This command will set up a non-scrolling display using mode CHAR4. To define this display another way, type:

DISPLAY  
24 CHAR4 GENLIST  
GENJVB

The DISPLAY command clears the memory and loads the vertical blank program. The 24 CHAR4 GENLIST specifies 24 mode lines of CHAR4. The generate jump vertical blank (GENJVB) command ends the display list. To define a multi-mode display, type:

DISPLAY  
12 CHAR4 GENLIST  
12 CHAR2 GENLIST  
GENJVB



To mix character and bit map graphics, try:

```
DISPLAY
96 BIT15 GENLIST
12 CHAR4 GENLIST
GENJVB
```

Now let's define a standard scrolling display. Type:

```
SCHAR4
```

This command is the same as:

```
DISPLAY
20 HORZ
8 VERT
48 CHAR4 VS HS GENLIST
GENJVB
```

The 20 HORZ command defines the width of the screen, which is 2 TV displays- single display is 10 units. The 8 VERT sets the number of scan lines in a graphic mode. The 48 CHAR4 GENLIST defines the height of the display in mode lines. Next let's define a multi-mode scrolling display. Type:

```
DISPLAY
30 HORZ
8 VERT
20 CHAR4 VS HS GENLIST
20 CHAR2 VS HS GENLIST
GENJVB
```

You can also define a vertical or horizontal display, only. Type:

```
DISPLAY
50 HORZ
24 CHAR4 HS GENLIST
GENJVB
```

or

```
DISPLAY
8 VERT
64 CHAR4 VS GENLIST
GENJVB
```

The HS command specifies which mode lines are to be scrolled. The VS command should always be used in a vertical scrolling display. Let's define a display which will only scroll a portion of the screen. Type:

```
DISPLAY
50 HORZ
8 CHAR4 HS GENLIST
8 CHAR4 GENLIST
8 CHAR4 HS GENLIST
GENJVB
```

The middle 8 mode lines will not scroll. If you wish to see the display list, type:

TLIST

To type a portion of the list, input the starting and ending mode line, as follows:

8 16 TLIST

If you like working with hex numbers, type:

```
HEX
TLIST
```

For decimal numbers, type:

DCX

A display list can be edited with the insert and delete

line commands. To delete mode line 8, type:

8 DLINE

To insert graphic mode CHAR4 at line 8, type:

8 CHAR4 ILINE

That covers most of the commands in the DEF-PIC module.

## " HOW TO " DRAW A PICTURE

Before leaving the DEF-PIC module, type:

SCHAR4

The DRAW-PIC module must be loaded to draw a picture, type:

DRAW-PIC

A menu should appear listing all the commands. To define a cursor with the PEN command, type:

### 1.1 PFO PEN

To set the screen, type:

DRAW

The screen should turn black, with the cursor in the left upper corner. To make sure the cursor is enabled, press:

START

Use the joystick to move the cursor. Pressing the trigger will mask in the pixels. If the screen resets to the text display typing out OPENING FONT, just wait a few seconds and the picture will return. To return to the menu screen,



press:

RESET

The cursor can be of different sizes and playfields. Try the following pens. Type:

4 16 PF1 PEN  
20 4 PF2 PEN  
20 16 PF0 PEN

Let's change the color of a playfield, type:

COLORS

The color names should be listed. For a listing of the register names, type:

NREG

To change the color of playfield 0, type:

BLUE 4 LUM COLPF0 REG

This will set PF0 to blue with a luminance of 4. The REG command will store the value. You can skip the LUM command and input the color directly, by typing:

255 COLPF0 REG

Let's define a cursor from the picture, type:

DRAW

Press:

SELECT  
M

The cursor frame should flash. To change the cursor size,

press the joystick and trigger at the same time. Release the trigger to move the frame. Move the frame around the desired pixels and, press:

START

A new cursor has been defined. To adjust the cursor in pixel increments, press:

SELECT

Move the frame to the new position, and press:

START

The START key always enables the cursor while the SELECT key performs operations on the cursor. Next, let's define a cursor with a transparent background. Press:

RESET

Type:

TRAN-ON

DRAW

Now define a cursor which is not a solid color, using the above method. The background will not be masked. To adjust the cursor movement, press:

SELECT

A

Now move the cursor frame half the cursor width and height, and press:

START

The cursors should now overlap. Let's try the outlined cursor next, press:

RESET

Type:

24 16 PFO PEN  
DRAW

Fill the bottom half of the screen with the solid color,  
then press:

RESET

Type:

OUT-ON  
DRAW

Now, define a new cursor using the SELECT key, with the  
pixels outlined by the background. Press:

START

Try drawing the new cursor in the area filled with the  
solid color. The cursor should be outlined with the  
background. The next step is to scroll the display. Press:

OPTION

The screen should scroll when the joystick is pressed. To  
increase the scrolling speed, press:

RESET

Type:

4 4 VSP  
4 4 HSP  
DRAW



The scrolling will increment 4 color clocks and scan lines at a time. Next press:

RESET

Type:

1 4 VSP

1 4 HSP

DRAW

The scrolling should be faster to the right and down. Next, let's try the zoom, press:

SELECT

Z

A large frame should appear. Move the frame to the desired position and press the trigger. The screen should be zoom with a small menu. Select a playfield and drawing mode by pressing the appropriate number. To draw a circle, use the HWAND or VWAND to define the radius. To expand or compress the picture, press one of the arrow keys ( pressing control is not needed ). To return to the main picture, press:

R

N

Let's try a different graphic mode, press:

RESET

Type:

DEF-PIC  
DCHAR6  
DRAW-PIC  
TRAN-OFF  
OUT-OFF  
8 8 PF0 PEN  
TEXT-ON  
DRAW

Press:

START

Fill in a portion of the screen with the cursor defined above. To input text, simply press any key. Text and graphics can be mixed anywhere on the screen. The graphics mode CHAR6 allows only two colors per character. To demonstrate this, press:

RESET

Type:

TEXT-OFF  
1 1 PF1 PEN  
DRAW

Now move the cursor where PF0 was drawn. You should notice that large blocks of pixels are changing colors. The cursor is changing the character name to PF1. Next, make a cursor using the SELECT key described above. This cursor will not change the character name, so the color of the cursor will be the color on the screen. There is one more type of cursor you can design- the character cursor, press:

SELECT  
C

The cursor will now move in character increments. Make a cursor with two colors, and press:

START

The character cursor will change the character name. This cursor will also limit the number of characters being generated. To make a cursor that will only change the character name, press:

RESET

Type:

8 8 PF2 PEN  
CPEN  
DRAW

The CPEN command will clear the pixel data, and will change the character name only. Next, zoom a portion of the screen. The zoom in graphic modes CHAR6 and CHAR7 will show only one color since the desired color is determined by the character name. The zoom is a bit map display. The next order of business is the GTIA modes. Press:

RESET

Type:

DEF-PIC  
DISPLAY  
20 HORZ  
24 CHAR2 HS GENLIST  
GENJVB  
DRAW-PIC

There are four commands (GTIA-OFF, GTIA-LUM, GTIA-REG, and GTIA-COL) which control the GTIA modes. Let's try the luminance mode, type:



```
GTIA-LUM  
BLUE 0 LUM COLBK REG  
1 1 8 PEN
```

Do not use the PFO commands when in the GTIA modes. To enable the shading cursor, type:

```
0 15 1 SHAD-ON  
DRAW
```

The cursor will now increment through all the luminances of blue. The 0 15 in the SHAD-ON command is the range of luminances. Press:

RESET

Type:

```
3 12 1 SHAD-ON  
DRAW
```

The luminance range is now 3 thru 12. The 1 in the SHAD-ON command is the increment, press:

RESET

Type:

```
0 15 -1 SHAD-ON  
DRAW
```

The shading will now start at the bright luminances. Let's try the color GTIA mode, press:

RESET

Type:

GTIA-COL  
BLACK 4 LUM COLBK REG  
DRAW

The shading cursor will now increment thru all the colors with the luminance of 4. To turn off the shading cursor, press:

RESET

Type:

SHAD-OFF  
DRAW

The cursor should be the last color drawn. The horizontal scrolling of a GTIA display must be in even increments. Press:

RESET

Type:

2 2 HSP  
DRAW

Press:

OPTION

Next, let's discuss the interrupt editor, press:

RESET

Type:

```
DEF-PIC
DCHAR4
DRAW-PIC
```

The interrupt editor is used to change a register while the TV screen is being drawn. The first step is to specify the mode line. Type:

```
12 EDINT
```

The next step is to insert the interrupt. Let's change the background color from black to green. Type:

```
0 COLBK GREEN 2 LUM ININT
```

After editing the interrupts, type:

```
GENINT
```

This command will generate the code for the interrupts. Type:

```
1 1 PFO PEN
DRAW
```

The upper half of the background should change from black to green. To change the color of a playfield, press:

```
RESET
```

Type:

```
12 EDINT
1 COLPFO BLACK 0 LUM ININT
GENINT
```

The 1 before COLPFO specifies the order in which the interrupts will occur. To find the mode line position of

the cursor, type:

CPOS

To list the interrupts, type:

TINT

DRAW

The cursor should change colors half way down the screen. Next, let's delete the interrupts. Press:

RESET

Type:

12 EDINT

COLBK DELINT

COLPFO DELINT

GENINT

We could have typed:

CLRINT

Which would clear all of the interrupts. Type:

DRAW

The screen should return to normal. That's most of the commands in the DRAW-PIC module.

## " HOW TO " SAVE A PICTURE

To save a picture the DISK module must be loaded. Type:

DISK

The disk menu should appear. Remove the Tiger Graphics disk, and insert a formatted disk. Type:

## SAVE-PIC

After you have saved the picture, reinsert the Tiger Graphics disk. The last step in drawing a picture is to compress and relocate the data so that you can call the picture from a BASIC, or ASSEMBLY language program. Type:

## COMPRESS

The program will ask you where you want the picture located. Locate the picture at the 8K border (BASIC). After the picture has been compressed, remove the Tiger Graphics disk and insert the formatted disk. Type:

## SAVE-CMP

The file name of a compressed picture should end with OBJ. We are now ready to test the picture from BASIC. Insert the BASIC cartridge and load DOS. Load the picture using the binary load command (L). Use the BASIC USR command to enable the picture. Type:

```
10 P=USR(38754)
20 GOTO 20
```

## RUN

To set the picture from an assembly program, type:

```
        JSR $9763
LOOP    JMP LOOP
```

This completes the " HOW TO " section.

## DISPLAY LIST

The graphic modes are determined by a display list. The display list is a set of instructions which determine the graphic mode for vertical strips of the TV screen. The graphic mode determines the size of the strip. Throughout this manual a vertical strip defined by a graphic mode will be called a mode line. A mode line can range from one to sixteen scan lines. There are 192 scan lines in a TV screen. The horizontal width of the TV screen is divided into 160 color clocks. The horizontal resolution is different multiples of the color clocks. Tiger Graphics uses the display list instruction, which will reload the display map address for each mode line.

## GRAPHIC MODES

The color of the pixels are defined by the color registers. There are five playfields and four player/missiles color registers. The pixel data is located in the display memory. The pixel data act as pointers to the color registers. The pointers can be either one, two, or four bits. When the pixel data is one bit, only two color registers can be addressed. Four color registers can be addressed when the pixel data is two bits. The GTIA mode uses four bits per pixel, which will either define the color or luminance directly, or will point to one of the nine color registers. All of the graphic modes available in the ATARI are variations in the pixel resolution of the above methods of addressing the color registers.

There are two ways to generate graphics; bit map, and character sets (fonts). In a bit map display, the pixel data is located in the display memory. In a character display, the pixel data is located in a character set. The terms character set and fonts mean the same thing. A font pointer is stored in the memory map. Either one or two bits of the font pointer, called the character name, can perform special operations on the character.

## GRAPHIC MODES

The graphic modes are divided into three categories, character map, bit map, and GTIA. Tiger Graphics uses all of the modes.

## CHARACTER MODES

There are either 128 or 64 characters in a font. The 128 character fonts are either multi-color or high resolution characters. The character name (1 bit) in the multi-color fonts select either the COLPF2 or COLPF3 color registers. The character name in the high resolution fonts can reflect, invert, and blink the character. The 64 character fonts are two color characters. The character name (2 bits) will select one of the color registers. COLBAK is always used for the off pixels.

## CHAR2, CHAR3

The pixel data in these modes is one bit which will point to either COLPF1 or COLPF2. The on pixels will use the color in COLPF2 and the luminance in COLPF1. The off pixels will use both color and luminance in COLPF2. Only one color and two luminances are allowed. The pixel resolution is 320H by 192V, the maximum the ATARI can generate. Since there are only 160 color clocks in a TV screen, artifacts will be generated when the pixels change during a color clock. An artifact is a pixel which has a different color than the value in the color register. CHAR3 is used to draw true lower case characters. Tiger Graphics can draw in CHAR3, but the cursor position will not be accurate. The character name can reflect, invert, and blink the character. ( See CHACTL )

## CHAR4, CHAR5

Two bits are used as the pointers which allow four colors per character. The character name will select either

COLPF2 or COLPF3 for the fourth color. This allows a maximum of five colors, but COLPF2 and COLPF3 cannot be displayed in the same character. CHAR4 has a vertical resolution of 192 pixels, while CHAR5 vertical resolution is 96 pixels. The horizontal resolution for both modes is 160 pixels.

#### CHAR6, CHAR7

One bit is used as the pointer. The off pixel will always use COLBK. The character name selects the color register of the on pixels. Four playfield registers can be selected, allowing five colors to be displayed. However, only two colors can be displayed in a character. CHAR6 has a vertical resolution of 192 pixels, while the CHAR7 vertical resolution is 96 pixels. The horizontal resolution for both modes is 160 pixels.

#### BIT MAP MODES

#### BIT8, BIT10, BIT13, BIT14

Two bits are used for the pointers. Only four colors are available in the bit map modes ( COLBK, COLPF0-2 ).

#### BIT9, BIT11, BIT12

One bit is used for the pointer. The on pixel will use the COLPF0 register, while the off pixel uses the COLBK register.

#### BIT15

This is the high resolution bit mode in which the on pixel uses the color of COLPF2 and the luminance of COLPF1. The off pixel uses both color and luminance of COLPF2.



## GTIA MODES

The GTIA modes are included in all the new ATARI computers. They are not new graphic modes, but a different interpretation of the existing modes. ATARI documentation states that GTIA modes will only work in mode BIT15 but examination of the hardware shows this is not true. The two graphic chips in the ATARI are called ANTIC and POKEY. The ANTIC chip decodes the different graphic modes and gets the data from memory ( DMA ). ANTIC passes the data to POKEY which converts it to a video signal. GTIA modes are enabled by setting the PRIOR register. PRIOR is located in POKEY, which is pass ANTIC in the data stream. This means that all graphic modes will be GTIA modes when enabled. For the cursor to work properly, use either BIT15 or CHAR2. TigerSoft will not guarantee this fact, but it seems to work. GTIA characters add a powerful feature to the graphic system.

There are three GTIA modes available which all have the same resolution (80H by 192V), and require four bits per pixel. To mix GTIA modes, the interrupt editor must be used to change the PRIOR register.

### GTIA-LUM

This is the luminance shading mode. The color is set by COLBK. The luminance is determined by the logical OR of the pixel data, with the luminance in COLBK. To get the full range of shading, set the luminance of COLBK to zero.

### GTIA-COL

This mode allows all sixteen colors to be displayed at once. The color is determined by the logical OR of the pixel data, and the color in COLBK. The luminance is set by the COLBK. The color in COLBK should be set to zero.

## GTIA-REG

This mode uses the pixel data as a pointer to one of the nine color registers. The color and luminance of the pixel is set by the color register. The cursor frame will not work properly in this mode.

## CHARACTER SETS ( FONTS )

The heart of Tiger Graphics is its ability to generate fonts. The main problem with font graphics is the limited number of characters in a font. Tiger Graphics solves this problem by moving characters from a filled font to either an existing font or to a new font.

Lets first examine how the fonts are organized. Two 64 character fonts make up a 128 character font. The last 16 characters in the 64 character fonts are allocated for the cursor. Initially the font is set-up as shown below.

0-47      GRAPHIC CHARACTERS

48-63     CURSOR CHARACTERS

64-111    GRAPHIC CHARACTERS

112-128   CURSOR CHARACTERS

The largest cursor can be 16 or 32 characters for 64 or 128 character fonts respectively. The cursor size can be decreased by using the CSIZE command. The smallest cursor allowed is 10 characters in the 64 character font or 20 characters in the 128 character font. The cursor size can only be decreased.

Opening a filled font is controlled by two limits; the number of freed characters in the filled font, and the total number of characters in the destination font. This can be a new or existing font. Tiger Graphics will search

different combinations of sections until these limits are satisfied. A section can contain between one and eight mode lines. After four fonts are opened, the program will first try to link the filled font to an existing font. If this fails, a new font will be opened.

If the limits freed and total characters cannot be satisfied, two error messages are outputted. CAN'T OPEN FONT means the filled font can't be opened to a new or existing font. The error message, ALL FONTS FILLED, means the filled font can't be linked to an existing font, and there is no more empty fonts available. The SETFT command can be used to change the free and total limits. First type the command to display the present limits. Decrease the free limit and increase the total to allow the filled font to be opened. The limits are set for a 64 character font, and are multiplied by two for the 128 character fonts. Let's look at an example of setting the limits.

First, to locate the start of the cursor characters use the FONTSZ command. The computer responses are on the right.

```
FONTSZ    48 OK
SETFT     15 33 OK
```

To set the free limit to 10 characters and the total to 38 characters, type:

```
10 38 SETFT    OK
```

The minimum free characters in the destination font will be 10 characters. This value is calculated by subtracting the total value from the FTSIZE ( 48-38 ). If the total value is set too high, the destination font can be filled.

## INTERRUPTS

An interrupt can be compared to the bell on a telephone. If the telephone did not have a bell, you would have to constantly pick up the telephone to see if someone was calling. An interrupt works the same way, it informs the computer to perform a certain operation at a desired time. Our main concern is the display list interrupt. This interrupt occurs at the beginning of a mode line. Most of the graphic registers can be changed by using the interrupts.

The display list interrupt is a very powerful feature in the ATARI because it allows you to change just about any graphic register, at any time. This feature combined with the many graphic modes and player/missile graphics, makes the ATARI graphic system the most powerful on the market.

The first rule about generating interrupt is that the changes do not take effect until one scan line later. Consecutive interrupts, in bit map modes with a vertical resolution of 192 pixels should not be used. The interrupts will also slow down the system, so try to make all the changes with the fewest number of interrupts possible.

The interrupt code was designed for speed, not memory efficiency. A typical interrupt looks like this:

```
PLA
STA WSYNC
LDA DATA
STA REGISTER
LDA DATA
STA REGISTER
JMP EXIT
```

Each interrupt has a vector stored in a table. The exit routine will get the next vector. The vector pointer is reset during the vertical blank interrupt.

## SCROLLING

Two methods (coarse and fine scroll), are used for both vertical and horizontal scrolling. For coarse vertical scrolling, a jump instruction is used to move the display up or down the list. The scan lines/mode will determine when the screen should coarse scroll. The horizontal coarse scroll is achieved by adding an offset to the display list. The coarse horizontal scroll is always 16 color clocks. The fine scroll is achieved by two hardware registers. These registers can shift the display in color clocks or scan line increments. The fine horizontal scroll can shift up to 16 color clocks, while the vertical fine scroll can shift the display up to 16 scan lines.

The horizontal display is divided into scroll units. Each unit is 16 color clocks wide. There are 10 scroll units in one standard screen. The minimum scroll width is 11 units. All other widths are in multiples of the scroll unit. It is possible to mix non-scrolling and horizontal scrolling mode lines on a screen. Tiger Graphics was designed to scroll character mode lines, but it is possible to scroll a few bit map lines. Too many bit map lines will cause the system to crash.

The vertical scrolling mode lines must all have the same scan lines/mode. This rule occurs since the coarse scroll is determined by the scan lines/mode of the mode line. Course vertical scroll will always move only one mode line at a time. An exception is if the mode line with a different scan lines/mode does not scroll out of the screen.

Example:

```
DISPLAY
8 VERT
12 CHAR4 VS GENLIST
4 CHAR5 VS GENLIST
12 CHAR4 VS GENLIST
GENJVB
```

The graphic mode CHAR5 has a different scan lines/mode, but will not scroll out of the display.

The interrupts will vertically scroll in sync with the display. When an interrupt scrolls in or out of the display, the shadow or hardware registers are changed corresponding to the interrupt changes.

The scrolling can be controlled from BASIC by either the joystick or an auto-scroll option. The auto-scroll will scroll the screen the desired number of times ( see VERTICAL BLANK ).

### SHADOW REGISTERS

Some of the register contents are destroyed when a TV screen is drawn. To solve this problem, shadow registers are used to update the hardware registers. Tiger Graphics will store the register data in a table located in the interrupt routine. When the picture is set, these values are moved to the shadow or hardware registers.

## MODULES

Tiger Graphics is divided into three modules. The module must be loaded to use their commands. To load a module, simply type its name. To abort module loading, press RESET.

## KERNAL COMMANDS

The KERNAL is loaded during booting and is always available.

## DISPLAY LIST INTERRUPT EDITOR

There are four commands used to generate an interrupt; EDINT, ININT, DELINT, and GENINT. The EDINT command will specify the mode line position of the interrupt. Once the mode line has been specified, the insert interrupt (ININT), or delete interrupt (DELINT), commands may be used. More than one register can be changed at an interrupt location. The order of register operation must be specified. After editing the interrupts, use GENINT to generate the object code. The TINT command will type the interrupt mode line, the registers changed, and the data addresses. The data can be changed using a POKE statement from BASIC. The data addresses are accurate after GENINT is used. Tiger Graphics allows 33 interrupts changing one register.

## INTERRUPT COMMANDS

### EDINT

INPUT- mode line

Edit interrupt specifies to which mode line you wish to insert or delete an interrupt.

### ININT

INPUT- order, register, data

Insert interrupt at the desired mode line specified by

EDINT. There is no limit to the number of register operations to be performed at a interrupt location. The order specifies which operation is to be performed first.  
Example:

```
0 HPOSPO 40 ININT
1 PRIOR 1 ININT
```

#### DELINT

INPUT- register

Deletes the selected register at the mode line specified by EDINT.

Example:

```
COLBK DELINT
```

#### CLRINT

INPUT- ...

Clears the interrupts except CHBASE.

#### GENINT

INPUT- ...

Generates the new code for the interrupts after an editing operation has been performed. This command must be used after any editing of the interrupts

#### TINT

INPUT- ( mode lines )

Types the interrupt list. The mode lines are optional. If no mode lines are specified, the entire list is typed.

### COLORS COMMANDS

#### LUM

INPUT-color, luminance

OUTPUT-color OR with luminance

Performs the logical OR of the color and luminance values into the proper format.

Example:

```
BLUE 2 LUM REG
GREEN 4 LUM ININT
```



## REG

INPUT- data, register name

Store data into the shadow register.

Example:

BLUE 3 LUM COLBK REG

55 HPOSPO REG

## COLORS

INPUT- ...

Types color names.

## NREG

INPUT- ...

Types graphic registers

## MODES

INPUT- ...

Types graphic modes.

## TREG

INPUT- ( name ) ( name )

Types shadow register values. All of the shadow registers are listed if no name is given. Typing one name will list the registers starting at the name. Typing two names will list the values between the names.

## PREG

INPUT- ...

Prints shadow register values.

## MENU-OFF

INPUT-

Turns off the menu listing.

## MENU-ON

INPUT- ...

Turns on the menu listing.

**MAKEBACK**

INPUT- ...

Makes a back-up copy of Tiger Graphics. Back-up copies can be made only from the Tiger Graphics disk. The Tiger Graphics disk should be inserted with no modules loaded before using this command.

## DEF-PIC MODULE COMMANDS

The DEF-PIC module is used to create and edit the display list, which defines the picture. Also included is a text editor for programming the display list, register changes, or interrupt changes.

The following is the graphic names used by Tiger Graphics. These commands should be used with the GENLIST command.

## Character modes

CHAR2 CHAR3 CHAR4  
CHAR5 CHAR6 CHAR7

## Bit map modes

BIT8 BIT9 BIT10  
BIT11 BIT12 BIT13  
BIT14 BIT15

The following commands will set a standard screen in the desired graphic mode.

DCHAR2 DCHAR3 DCHAR4 DCHAR5  
DCHAR6 DCHAR7  
DBIT8 DBIT9 DBIT10 DBIT11  
DBIT12 DBIT13 DBIT14 DBIT15

The following commands will set a scrolling display two screens in width, and height.

SCHAR2 SCHAR4 SCHAR5  
SCHAR6 SCHAR7

## DISPLAY

INPUT- ...

Loads the vertical blank code, and clears the display map and fonts. It will clear any existing pictures. This is the first command used in defining the display.

**GENLIST**

INPUT- # lines, mode, ( VS, HS )

Generates display list; may be used repeatedly to set up multiple-mode displays. The HS command is optional. The graphic mode names, and ANTIC number are interchangeable. A maximum of 252 mode lines are allowed.

Example:

38 CHAR4 VS HS GENLIST  
24 2 GENLIST or 24 CHAR2 GENLIST

**GENJVB**

INPUT- ...

Generates jump vertical blank; must be used to end the display list.

**TLIST**

INPUT- ( start, end )

Types display list; will either type the entire list, or a part of the list.

**PLIST**

INPUT- ( start, end )

Prints display list. See TLIST.

**ILINE**

INPUT- mode line, mode, ( VS, HS )

Inserts line; will adjust the display list and memory map to insert a line at the desired position.

Example:

8 CHAR4 VS HS ILINE

**DLINE**

INPUT- mode line

Delete line; will adjust the display list and memory map to delete a line at the desired position.

## SFILL

INPUT- ( data )

Fills the screen with the desired data. If nothing is inputted, will fill the screen with the number 255.

## TFONTS

INPUT- ...

Types the font base addresses.

## PFONTS

INPUT- ...

Prints the font base addresses.

## TEXT EDITOR and FORTH

Included in Tiger Graphics is a text editor which can be used to program displays. In FORTH, a screen refers to a block memory 512 bytes, which is used to store text data or source code. The screens are stored on the disk. Each screens is numbered for loading from the disk. To enable the text editor, type EDITOR. To edit a screen, enter the screen number followed by L. The minus key will place the cursor at line 0. The control arrow keys will move the cursor. To leave a screen, press the ESC key twice. After a screen has been edited, use the FLUSH command to store the screen on disk. Screens 1 thru 34 are available to the user. The screen contains source code which can be either compiled or interpreted. Screens 1 thru 10 contain examples of how to use the text editor.

## EDITOR

INPUT- ...

Enables the editor vocabulary. Use this command before editing a screen, or after pressing reset.

## L

INPUT- screen number

Edits a screen.

## FLUSH

INPUT- ...

Stores screen on the disk.

## EMPTY-BUFFERS

INPUT- ...

Clears the disk buffers. Will not affect the screens on the disk.

## CLEAR

INPUT- screen number

Clears screen in disk buffer. Use the FLUSH command to store the cleared screen on the disk.

:

INPUT- ...

Colon definition, use to define a word in the FORTH dictionary.

;

INPUT- ...

Semi-colon closes the word definition.

Example;

: DOG ." DOGS ARE NEAT " ;

DOG is compiled in the dictionary, and when executed will type out DOGS ARE NEAT.

: ADD 2 2 + . ;

ADD will perform the addition of 2 + 2 and print the results.

INPUT- number

Prints out the top of the stack in signed integer form.

U.

INPUT- number

Prints out the top of the stack in unsigned integer form.

**DO..LOOP**

INPUT- limit, index

Defines a loop. This command will only work inside a colon definitions.

Example:

```
: TEST 12 0 DO 2 2 + . LOOP ;
```

TEST will execute 2 2 + . 12 times.

**LOAD**

INPUT- screen #

Compiles the source code in the screen.

## DRAW PICTURE MODULE COMMANDS

The cursor will work the same for bit map and character modes except the character modes has a preview feature. The preview will always mask in the cursor. The old pixels will be returned if the trigger is not pressed. The cursor frame uses the missiles 2 and 3. To change the color of the frame, use COLPM2 and COLPM3.

## FUNCTION KEYS

## START

The START key will activate the cursor. The cursor cannot cross graphic modes when it is activated. If the cursor currently loaded does not match the graphic mode, an error bell will sound and the SELECT key will be activated.

## SELECT

The SELECT key performs operations on the cursor. After pressing the select key, the cursor can be moved in pixel or character increments. The movement is unbonded by the graphic modes. An error bell will occur, if the graphic mode of the cursor and its new position do not match.

There are four keys used in the SELECT mode that perform the operations described below.

## M - MAKE CURSOR

To design a new cursor move the frame to the new position and press key M. If the current cursor size will not fit into the new position, it will reset to the smallest size. To adjust the cursor size, press the trigger and joystick at the same time. A cursor generated by this method will not affect the character names.



## C - CHARACTER CURSOR

The character cursor can only be used in the character graphics modes. The size of the cursor will be in increments of characters, not pixels. This cursor will change the character names. Using the character cursor will limit the number of characters being generated.

## A - ADJUST CURSOR MOVEMENT

The amount of pixels or characters the cursor is to be moved can be determined by moving the cursor frame to the next desired position.

## Z - ZOOM

To zoom a portion of the screen, move the cursor frame to the desired location and press the Z key. A large frame will appear which can be moved by the joystick. Different graphic modes cannot be zoomed together. When the frame is in the desired location, press the trigger. A menu of the zoom drawing modes will appear.

BAK, PF0, PF1, PF2, PF3

Sets the cursor to the desired playfield. PF3 is not available. The playfield can be changed at any time.

PLOT

Plots a single pixel of the selected playfield.

DRAWTO

Draws a line between the two cursor positions.

HWAND

The horizontal wand will lock in the horizontal position of one cursor and will draw a horizontal line to the other cursor position. The wand can be moved up or down.

**VWAND**

The vertical wand works the same as HWAND except that the vertical position is locked and vertical lines are drawn.

**CIRCLE**

Use the VWAND or HWAND to define the radius of the circle. Half or quarter circles can be drawn by moving one of the cursors to the edge of the screen.

**CIRCLE-OUT**

Works the same as circle, but will only outline the circle.

**COMPRESS or EXPAND PICTURE**

These commands, which are not shown on the menu, will expand or compress the picture when one of the arrow keys is pressed ( pressing control is not needed ). The picture is always expanded to the right or down, and compressed to the left or up.

**RETURN**

Returns to the main picture, with the option of restoring the old picture.

There are two types of zooms, A and B. The DRAWTO command in zoom B is an approximation for modes CHAR4, CHAR6, BIT12, and BIT14, which may produce double lines.

GRAPHIC MODE	MAGNIFICATION ( HxV )	
	A	B
CHAR2	4x4	same
CHAR3	4x4	same
CHAR4	2x4	4x4
CHAR5	4x4	same
CHAR6	2x4	4x4
CHAR7	2x2	same
BIT8	1x1	same
BIT9	1x1	same
BIT10	1x1	same
BIT11	2x2	same
BIT12	2x4	4x4
BIT13	2x2	same
BIT14	2x4	4x4
BIT15	4x4	same

When returning from zoom, the select key is enabled. The old cursor is destroyed and a new one must be made or the zoom can be used again. The zoom will only display PF0 in graphic modes CHAR6 and CHAR7. GTIA displays cannot be zoomed.

### OPTION

The OPTION key will enable scrolling. When scrolling is disabled, the screen will automatically reset to the nearest scroll border. The cursor will reset to the center of the screen at a character or byte border after scrolling. If the graphic modes of the cursor and position do not match, the select key will be enabled. If the display is not a scrolling screen, the OPTION key will reset the cursor to a character or byte border.

## DRAW

INPUT- ...

Draw picture; will set the screen to the current picture loaded.

## PEN

INPUT- x,y,playfield

Make pen; will set up the cursor at the desired size and color register. The playfield commands are listed below. For GTIA pens, use either the color name or the luminance value.

Playfield commands:

COLBK- BAK	COLPF2- PF2
COLPF0- PF0	COLPF3- PF3
COLPF1- PF1	

CHAR2 playfield commands.

CURSOR		CHAR
PLAYFIELD NAME		
PF0	PF1	1
PF1	PF1	0
PF2	PF2	0
PF3	PF2	1

GTIA-REG playfield commands.

0-COLPM0	4-COLPF0	8-COLBK
1-COLPM1	5-COLPF1	
2-COLPM2	6-COLPF2	
3-COLPM3	7-COLPF3	

Example: 4 4 PF0 PEN  
 2 8 BLUE PEN ( GTIA-COL )  
 3 4 0 PEN ( GTIA-LUM )

## CPEN

INPUT- ...

Clears the cursor data. This command is useful to change the character names only.

## VSP

INPUT- up, down

Sets the vertical scrolling speed.

## HSP

INPUT- left, right

Sets the horizontal scrolling speed.

## TRAN-ON

INPUT- ...

Sets the transparent background when making a cursor.

## TRAN-OFF

INPUT- ...

Turns off the transparent background.

## OUT-ON

INPUT- ...

Outlines the cursor with the background when making a cursor. The transparent background must be set for this command to work.

## OUT-OFF

INPUT- ...

Turns off the outline.

## TEXT-ON

INPUT- ...

Enables text to be inputted at the current cursor location. Press any key when in the START mode.

## TEXT-OFF

INPUT- ...

Disables the text mode.

## CSIZE

INPUT- ...

Decreases the number of characters allocated for the cursor in the character modes. The cursor can never be increased. Use CSIZE when a filled font cannot be opened.

## SETFT

INPUT- ( MINFREE, MAXCHAR )

Sets the desired limits when opening a filled font ( see FONTS ). When no values are input, the present values are typed.

## FTSIZE

INPUT- ...

Types the start of the cursor characters.

## CPOS

INPUT- ...

Types the current mode line position of the cursor.

## AND-ON

INPUT- ...

Will logically AND the two pixels when compressing the picture in the zoom mode.

## OR-ON

INPUT- ...

Will logically OR the two pixels when compressing the picture in the zoom mode.

## ZOOM-A

INPUT- ...

Sets ZOOM table A.

## ZOOM-B

INPUT- ...

Sets ZOOM table B.

## GTIA COMMANDS

The GTIA commands have two things in common. They will store the proper value in the PRIOR register, and will interpret the cursor as a GTIA cursor when graphic mode CHAR2 or BIT15 is used.

## GTIA-OFF

INPUT- ...

GTIA off; will store a one in the PRIOR register, and turn off the GTIA cursor.

## GTIA-LUM

INPUT- ...

GTIA luminance; will store a hex 41 in the PRIOR register, and turn on the GTIA cursor.

## GTIA-COL

INPUT- ...

GTIA colors; will store a hex C1 in the PRIOR register, and turn on the GTIA cursor.

## GTIA-REG

INPUT- ...

GTIA registers; will store a hex 81 in the PRIOR register, and turn on the GTIA cursor.

## SHAD-ON

INPUT- low, high, increment

Shading on, will increment or decrement the cursor each time the cursor is enabled ( trigger pressed ). The shading can range between 0 and 15. The increment can be positive or negative, and has a range between 0 and 15. If GTIA-COL or GTIA-REG is enabled, the cursor will increment thru all the colors or registers.

## SHAD-OFF

INPUT- ...

Shading off, will turn off the shading cursor.

## DISK MODULE

The DISK module will allow you to save the pictures on the disk. Tiger Graphics will generate binary files which are compatible with ATARI DOS. There are two save commands; SAVE-PIC, and SAVE-CMP. The SAVE-PIC command will save pictures which are only compatible with Tiger Graphics. Back-up are generated with existing file names. The SAVE-CMP command will save the picture for later use from BASIC. No pictures can be saved on the Tiger Graphics disk.

## LOAD-PIC

INPUT- ...

Load a picture from disk. The picture should not be compressed.

## SAVE-PIC

INPUT- ...

Save a picture for Tiger Graphics use only. Will change old file of the same name to BAK.

## SAVE-CMP

INPUT- ...

Save picture for end user (BASIC).

## DEL-PIC

INPUT- ...

Deletes picture.

## DIR

INPUT- ...

Lists disk directory.

## COMPRESS

INPUT- ...

This command will attempt to compress the fonts and move the display map just below the fonts. Three memory locations are available for the compressed picture. The



picture is always placed below the standard text screen location with either no cartridge, 8K, or 16K cartridge. The current BASIC and assembler/editor cartridges are 8K. (see MEMORY MAP) Tiger Graphics cannot work on a compress picture.

## GRAPHIC REGISTERS

The graphic register names are the same as in the **HARDWARE MANUAL**. The shadow names, if any, are listed below the description.

## COLOR REGISTERS

## COLBK

Background color  
(COLOR4)

## COLPF0-COLPF3

Playfield 0 color  
(COLOR0-COLOR3)

## COLPM0-COLPM3

Player/missile 0 color  
(PCOLOR0-PCOLOR3)

## GENERAL GRAPHIC REGISTERS

## DMACTL

Direct memory access (DMA) control  
(SDMCTL)

Bit#	Value	purpose
7,6	--	not used
5	1	enable DMA
4	1	1 line player/missile resolution
	0	2 line player/missile resolution
3	1	enable player DMA
2	1	enable missile DMA
1,0	0 0	no playfield
	0 1	narrow playfield (128 color clocks)
	1 0	standard playfield (160 color clocks)
	1 1	wide playfield (192 color clocks)

## GRCTL

Graphics control

Bit#	Value	Purpose
3-7	---	not used
2	1	enable latches on TRIG0-TRIG3.
1	1	enable player DMA to player graphic registers
0	1	enable missile DMA to missile graphic registers

## CHACTL

Character control, will perform special funtions on the characters in CHAR2 only.  
(CHACTL)

Bit#	Value	Purpose
3-7	--	not used
2	1	vertical reflect bit. This bit is sampled at the beginning of each mode line. If set, will reflect (invert) the characters (upside down characters)
1	1	Video invert flag. Will invert the colors of CHAR2 characters, if the character name is set to one.
0	1	Blank flag. Will blank the characters in CHAR2, if the character name is set to one. Setting and resetting this bit will blink the character.

## DLISTL

Store the low address byte in the display list counter.  
(SDLSTL)

## DLISTH

Store the high address byte in the display list counter.  
(SDLSTH)

## PRIOR

Stores the player priority and enables the GTIA modes.

(GPRIOR)

Bit#	Value	Purpose
7,6	0 0	disable GTIA
	0 1	enable GTIA luminance mode
	1 0	enable GTIA register mode
	1 1	enable GTIA color mode
4	1	multiple color player enable. This bit causes the logical OR of the colors of player 0 with player 1 and player 2 with player 3, when they overlap.
5	1	enable fifth player. Setting this bit allows the missiles to be used as a player with the color in COLPF3.
0-3	---	priority select, see below.

## Bit number

B3=1	B2=1	B1=1	B0=1
PF0	PF0	P0	P0
PF1	PF1	P1	P1
P0	PF2	PF0	P2
P1	PF3,P5	PF1	P3
P2	P0	PF2	PF0
P3	P1	PF3,P5	PF1
PF2	P2	P2	PF2
PF3,P5	P3	P3	PF3,P5
BAK	BAK	BAK	BAK

## CHBASE

Stores the high address byte into the character address base. The lower two bits are assumed to be zero.

(CHBAS)

## PMBASE

Stores the high address byte into the player/missile address base.

**HSCROL**

Shifts the display n color clocks to the right.

**VSCROL**

Scrolls the display n scan lines up.

**VDELAY**

Vertical delay; is used to give 2 line resolution player/missiles the vertical postioning of 1 line resolution. Setting a bit in VDELAY moves the object down one scan line.

**Bit#**

7	6	5	4	3	2	1	0
P3	P2	P1	P0	M3	M2	M1	M0

**WSYNC**

Wait for horizontal blank. Storing any data to this register will halt the microprocessor until the next scan line.

**PLAYER/MISSILE GRAPHIC REGISTERS****HPOSP0-HPOSP3**

Stores the horizontal position of the players. Hex 30 is the left edge, and hex D0 is the right edge of a standard screen.

**HPOSM0-HPOSM3**

Stores the horizontal position of the missiles.  
(see HPOSP0-HPOSP3)

## SIZEP0-SIZEP3

Player size control

Bit#	Value	Purpose
2-7	---	not used
1,0	0 0	normal size (8 color clocks)
	0 1	twice normal size (16 color clocks)
	1 0	normal size
	1 1	four times normal size (32 color clocks)

## SIZEM

Missile size control (see SIZEP0)

Bit#							
7	6	5	4	3	2	1	0
M3	M2	M1	M0				

## GRAFP0-GRAFP3

Player graphic register. The players data may be stored directly into these registers when player DMA is disabled (GRACTL). These registers are automatically loaded when player DMA is enabled.

## GRAFM

Missile graphic register. (see GRAFP0)

Bit#							
7	6	5	4	3	2	1	0
M3	M2	M1	M0				

## COLOR NAMES

GRAY, GOLD, ORANGE, RED-ORANGE, PINK,  
 PURPLE, PURPLE-BLUE, BLUE, ABLUE,  
 L-BLUE, TURQUOISE, GREEN-BLUE,  
 GREEN, YELLOW-GREEN, ORANGE-GREEN,  
 L-ORANGE, BLACK.

## VERTICAL BLANK ADDRESSES

The vertical blank addresses listed below are for no cartridge, 8K (BASIC) cartridge, or 16K cartridge. The number of bytes is shown in parentheses.

SCRLSEL (1) \$B81B \$981B \$781B  
47131 38939 30747

Scrolling select. Store a zero for joystick scrolling. Store a one for continuous scrolling.

SCRLDIR (1) \$B820 \$9820 \$7820  
47136 38944 30752

Controls the scrolling direction for continuous scrolling. The direction bits are the same as the joystick inputs. The upper four bits should be set to 1.

## BIT#

7	6	5	4	3	2	1	0	
1	1	1	1	1	1	1	1	NO SCROLLING
1	1	1	1	1	1	1	0	SCROLL UP
1	1	1	1	1	1	0	1	SCROLL DOWN
1	1	1	1	1	0	1	1	SCROLL LEFT
1	1	1	1	0	1	1	1	SCROLL RIGHT

VERTCNT (2) \$B81C \$981C \$781C  
47132 38940 30748

Continuous vertical scrolling count. Store the number of times the screen is to vertically scroll. Scrolling will stop when it reaches zero. VERTCNT is decremented during the VBLANK interrupt.

HORZCNT (2) \$B81E \$981E \$781E  
47134 38942 30750

Continuous horizontal scrolling count (see VERTCNT).

UPSPEED (1) \$B825 \$9825 \$7825  
47141 38949 30750

Store a value to set the up vertical scrolling speed in scan line increments.

DNSPEED (1) \$B826 \$9826 \$7826  
47142 38950 30758

Store a value to set the down vertical scrolling speed in scan line increments.

RTSPEED (1) \$B827 \$9827 \$7827  
47143 38951 30759

Store a value to set the right scrolling speed in color clock increments.

LTSPEED (1) \$B828 \$9828 \$7828  
47144 38952 30760

Store a value to set the left scrolling speed in color clock increments.

SETBAS \$B762 \$9762 \$7762  
46946 38754 30563

USR statement to this address will set the picture.

SETASM \$B763 \$9763 \$7763  
46947 38755 30564

A JSR statement to this address will set the picture.



## HELPFUL HINTS

Tiger Graphics eliminates most of the problems drawing in character modes except for one, the limited amount of characters. If you are not careful, all the fonts can be filled quickly. To prevent this, use the character cursor whenever possible. Let's look at a example of drawing a small tree. If the tree cursor is not in character increments, everytime the tree is drawn, new characters are made. The tree is roaming inside of a block of characters. To design a cursor in character increments using the M key, first set the size using the character cursor (C). Once the size is determined, remake the cursor using the M key.

The limited number of interrupts can cause a problem when a filled font is opened. Insert some dummy interrupts using the WSYNC register. If the interrupt memory is filled, use the command CLRINT, which will clear all the interrupts and regenerate the font interrupts.

There is a good change the beginnner will be confused after reading this manual. To fully understand Tiger Graphics, you must understand the graphic hardware. The beginner should not worry about all the graphic modes, most are useless. Use the modes CHAR4 or BIT14, which allows unrestricted use of four color registers. To see how the register work, experiment using the interrupt editor. The best way to understand the hardware is to change a register and look at the results.

## MEMORY MAP

The memory map for a compressed picture with no cartridge, 8K cartridge (BASIC), or 16K cartridge. All numbers are in hex values.

TEXT SCREEN	C000	A000	8000
	BC20	9C20	7C20

VERTICAL BLANK	BC1F	9C1F	7C1F
	B300	9300	7300

DISPLAY LIST	B2FF	92FF	72FF
	B000	9000	7000

	AFFF	8FFF	6FFF
--	------	------	------

FONTS

DISPLAY MAP

DISPLAY START (MODE LINE 0)

GRAPHIC MODES TABLE									
GRAPHIC MODE	BASIC MODE	ANTIC MODE	RESOLUTION HORZ	MODE/ VERT	SCAN/ SCREEN MODE	PLAYFIELDS BAK	PF0	PF1	PF2 PF3
CHAR2	0	2	320	192	24	8		X	X
CHAR3	-	3	320	192	19	10		X	X
CHAR4	-	4	160	192	24	8	X	X	X
CHAR5	-	5	160	96	12	16	X	X	X
CHAR6	1	6	160	192	24	8	X	X	X
CHAR7	2	7	160	96	12	16	X	X	X
BIT8	3	8	40	24	24	8	X	X	X
BIT9	4	9	80	48	48	4	X	X	
BIT10	5	10	80	48	48	4	X	X	X
BIT11	6	11	160	96	96	2	X	X	
BIT12	-	12	160	192	192	1	X	X	
BIT13	7	13	160	96	96	2	X	X	X
BIT14	-	14	160	192	192	1	X	X	X
BIT15	8	15	320	192	192	1	X	X	X











# TIGER GRAPHICS

Unleash the graphic power of the ATARI computer with TIGER GRAPHICS. Assembly language is no longer needed for scrolling and display list interrupts. Drawing pictures in character modes is the same as bit modes since TIGER GRAPHICS handles all the details of generating character sets. Written in FORTH. Requires 48K.

- **Custom Display List Editor**

Create a display list that uses all the graphic modes including GTIA in one TV screen.

- **Automatic Character Set Generation**

Generating graphics in the character modes has never been easier. TIGER GRAPHICS will generate and search for duplicate characters directly from the cursor. New character sets are opened when a set is filled, up to a maximum of sixteen character sets. The best feature of all is that no knowledge of character sets is needed.

- **Display List Interrupt Editor**

TIGER GRAPHICS can change all the graphic registers using the display list interrupt. Two easy to use commands can create a display using all the colors, mix GTIA modes, or change player/missile registers with the interrupt editor.

- **Scrolling**

With TIGER GRAPHICS you can create a scrolling display up to 40 TV screens in size. Mix scrolling with non-scrolling together for fantastic special effect displays.

- **Cursors**

The TIGER GRAPHICS cursor is adjustable in size and movement. The cursor can be multi-colored, outlined, with the option of a transparent background. A special shading cursor gives you added versatility for GTIA modes.

- **Zoom**

Zoom in on a portion of the screen and use the DRAWTO and TIGER GRAPHICS horizontal and vertical drawing wands. Expand or compress the picture with one easy key stroke.

## TIGER SOFT

91 Pocatello Trail, Rochester, N.Y. 14467